

CLAIMS

1
2
3 1. A method for capturing the contents of the files and directories in a
4 file system, said file system comprising a set of storage blocks in a mass storage system
5 including steps for

6 recording an active map in said file system of said storage blocks not avail-
7 able for writing data;

8 recording a consistency point in said file system including a consistent ver-
9 sion of said file system at a previous time, said consistency point including a copy of said
10 active map at said previous time; and

11 refraining from writing data to storage blocks in response to said active
12 map, and at least one of said copy of said active map at said previous time.

13
14 2. A method as in claim 1, wherein said step for refraining includes
15 determining a logical union of said storage blocks used by one or more of said copies of
16 said active map at said previous time.

17
18 3. A method as in claim 1, wherein said step for refraining includes
19 determining a subset of said storage blocks used by one or more of said copies of said
20 active map at said previous time.

1 4. A method as in claim 1, wherein said file system is a WAFL file
2 system.

3
4 5. A method as in claim 1, wherein said active map at said previous
5 time is a snapmap.

6
7 6. A method as in claim 1 and 5, including removing a root inode of
8 said snapmap using a snap delete.

9
10 7. A method as in claim 6, including steps for determining not to write
11 to a block after said step, provided the previous or next snapmap uses said block.

12
13 8. A method as in claim 1, including a copy-on-write mechanism for
14 copying modified data to a new block and saving old data in a current data block.

15
16 9. A method for capturing the contents of the files and directories in a
17 file system, said file system comprising a set of storage blocks in a mass storage system
18 including

19 recording a consistency point in said file system including a consistent ver-
20 sion of said file system at a previous time, said consistency point including a copy of said
21 active map at said previous time; and

1 returning to said file system at a previous time using said consistent version
2 of said file system following an unintended deletion or modification.

3
4 10. A method as in claim 9, wherein said consistent version includes a
5 pointer to a previous root block of the inode file.

6
7 11. A method as in claim 9, wherein said file system is a WAFL file
8 system.

9
10 12. A method as in claim 9, wherein said active map at said previous
11 time is a snapmap.

12 13. A method as in claim 9 and 12, including a snapdelete method for
13 removing a root inode of said snapmap.

14
15 14. A method as in claim 13, including steps for determining not to write
16 to a block after said snapdelete method provided a previous or next snapmap uses said
17 block.

18
19 15. A method as in claim 9, including a copy-on-write mechanism for
20 copying modified data to a new block and saving old data in a current data block.

1 16. A method for saving previous versions of an active file system in-
2 cluding the contents of the files and directories in a file system, said file system com-
3 prising a set of storage blocks in a mass storage system including steps for
4 writing modified files to unused data blocks;
5 keeping previous files in currently occupied blocks; and
6 recording a consistency point in said file system including a consistent ver-
7 sion of said file system at a previous time, said consistency point including a copy of said
8 active map at said previous time;

9
10 17. A method as in claim 16, including retrieving said file system at a
11 previous time using a pointer.

12
13 18. A method as in claim 16, wherein said pointer corresponds to a root
14 block of said file system at a previous time.

15
16 19. A method as in claim 16, wherein said file system is a WAFL file
17 system.

18
19 20. A method as in claim 16, wherein said active map at said previous
20 time is a snapmap.

1 21. A method as in claim 16 and 20, including a snapdelete method for
2 removing a root inode of said snapmap.

3
4 22. A method as in claim 20, including not writing to a block after said
5 snapdelete method provided a previous or next snapmap uses said block.

6
7 23. A method as in claim 16, including a copy-on-write mechanism for
8 copying modified data to a new block and saving old data in a current data block..

9
10 24. A method of operating a file system, said file system including an
11 active map of information indicating in-use and free blocks, said file system maintaining
12 a set of snapshots, each snapshot including a representation of said file system as it was at
13 an earlier time, said method including

14 making write allocation decisions in response to a copy of an earlier active
15 map included in at least one of said snapshots.

16
17 25. A method of operating a file system, said file system including an
18 active map of information indicating in-use and free blocks, said file system maintaining
19 a set of snapshots, each snapshot including a representation of said file system as it was at
20 an earlier time, said method including

21 computing a summary map in response to at least one copy of an earlier ac-
22 tive map included in at least one of said snapshots.

1
2 26. A method as in claim 25, including
3 making write allocation decisions in response to said summary map.
4

5 27. A method as in claim 25, wherein
6 said set of snapshots includes at least two said snapshots; and
7 a result of said computing includes an indicator of a union of all blocks in-
8 dicated by at least two said copies of earlier active maps included in said set of snapshots.

9
10 28. A method as in claim 25, wherein
11 said set of snapshots includes at least two said snapshots; and
12 said computing includes performing a bitwise logical operation on at least
13 two said copies of earlier active maps included in said set of snapshots.
14

15 29. A method as in claim 25, including
16 making write allocation decisions both in response to a current active map
17 and in response to said summary map.
18

19 30. A method as in claim 25, including
20 computing a combination of a current active map and said summary map;
21 and
22 making write allocation decisions in response to a result of said computing.

1
2 31. A method as in claim 25, including, for a selected portion of said
3 summary map
4 identifying a set of snapshots created since a recent update of said selected
5 portion; and
6 updating said selected portion in response to only a most recent one of said
7 snapshots.

8
9 32. In a file system including an active map of information indicating in-
10 use and free blocks, said file system maintaining a set of snapshots, each snapshot in-
11 cluding a representation of said file system as it was at an earlier time, said file system
12 maintaining a summary map in response to at least one copy of an earlier active map in-
13 cluded in at least one of said snapshots, a method of updating said summary map, said
14 method including

15 receiving a request to delete a selected snapshot;
16 for a block used by said selected snapshot, indicating said block is free in
17 said summary map only in response to a snapshot just prior to said selected snapshot and
18 in response to a snapshot just after said selected snapshot.

19
20 33. A method as in claim 32, wherein said indicating frees said block
21 only when both

1 said block is unused by said snapshot just prior to said selected snapshot;

2 and

3 said block is unused by said snapshot just after said selected snapshot.

4
5 34. A method as in claim 32, wherein said snapshot just after said se-
6 lected snapshot corresponds to an active file system.

7
8 35. In a file system including an active map of information indicating in-
9 use and free blocks, said file system maintaining a set of snapshots, each snapshot in-
10 cluding a representation of said file system as it was at an earlier time, said file system
11 maintaining a summary map in response to at least one copy of an earlier active map in-
12 cluded in at least one of said snapshots, a method of updating said summary map, said
13 method including

14 selecting a set of blocks maintained by said file system for which to per-
15 form a write allocation operation;

16 updating only a portion of said summary map corresponding to said set of
17 blocks, in response to said selecting; and

18 performing said write allocation operation in response to said updated
19 summary map.

20
21 36. In a file system including an active map of information indicating in-
22 use and free blocks, and said file system maintaining a set of snapshots, each snapshot in-

1 cluding a representation of said file system as it was at an earlier time, said file system
2 maintaining a summary map in response to at least one copy of an earlier active map in-
3 cluded in at least one of said snapshots, a method of updating said summary map, said
4 method including

5 while generating a consistency point, selecting a set of blocks maintained
6 by said file system and updating only a portion of said summary map corresponding to
7 said set of blocks.

8
9 37. In a file system including an active map of information indicating in-
10 use and free blocks, and said file system maintaining a set of snapshots, each snapshot in-
11 cluding a representation of said file system as it was at an earlier time, said file system
12 maintaining a summary map in response to at least one copy of an earlier active map in-
13 cluded in at least one of said snapshots, a method of updating said summary map, said
14 method including

15 refraining from indicating a selected block as free in response to whether
16 said selected block is included in said copy of an earlier active map.

17
18 38. In a file system including an active map of information indicating in-
19 use and free blocks, a method of updating said active map, said method including

20 maintaining a plurality of copies of said active map, at least a first said
21 copy being a substantially true representation of in-use and free blocks, and at least a sec-

5

6

7

8

[illegible]